11. Nikita

Program Name: Nikita.java Input File: nikita.dat

Nikita's friend Stelios has been researching efficient ways to represent connections within graphs, and has developed a shortest path matrix that shows this. The resulting chart contains the least number of steps it takes to get from one node to any other node. Nikita wants to take Stelios' work and do some analysis about the general nature of any particular graph, especially regarding the accessibility of any node to all other nodes within the graph. For example, he is interested in how many nodes directly connect with each node in the graph, what he considers the "degree" of a node, and generally how "close" each node is to the rest of the graph.

```
A - B  -  C
     \   /
       D
```

	A	B	C	D
A	0	1	2	2
B	1	0	1	1
C	2	1	0	1
D	2	1	1	0

	1	2
A	1	2
B	3	0
C	2	1
D	2	1

Nikita looks at one of the graphs Stelios has worked with, specifically the four-node graph shown above. The connection matrix in the center shows values that represent how many steps it takes to get to each node from any other node. He observes that the longest path in this situation is 2, and so develops a two-column "length tally" matrix to represent this situation, like the one on the right. This shows that the A node has **one** 1-link connection, and **two** 2-link connections. The B node has **three** 1-link connections, and C and D each have **two** 1-link and **one** 2-link connections.

Just by looking at the graph, Nikita can intuitively judge that the A node seems to be the most isolated, and that B is clearly the most centralized. He thinks about how to mathematically determine this, and comes up with two ways to measure this. First, he wants to measure the "degree" of each node. The formula he comes up with is this:

- Divide the number of 1-link connections for each node by one less than the number of nodes.

Using the length tally matrix shown above, the A node calculation would be 1/3, or 0.33. The B node would be 3/3, or 1.00. The C and D nodes both would be 2/3, or 0.67. Clearly the node with the least degree is A, and the one with the most degree was B, which supports the Nikita's original intuitive observation.

The "closeness" idea is a little more of a challenge, but after pondering a while he decides to use this formula. For each node:

- Find the sum of the number of 1-link connections times 1, 2-link connections times 2, 3-links * 3, etc.
- Divide that sum **into** the value representing one less than the total number of nodes in the graph.

For node A, the "closeness" calculation would be **3 divided by (1*1 + 2*2), or 3/5, or 0.60.**
For node B it would be **3 / (3*1 + 2*0), or 3/3, which is 1.00.**
Nodes C and D both have a "closeness" result of **0.67**.

This "closeness" measure at first seems to show the same result as the "degree", in that the A node appears to be the most remote node, with the least "closeness" value, and that B is the most central node, with the greatest "closeness" value, which leaves Nikita unsatisfied. He decides to look at one more example (see next page).

Nikita - cont

```
A - B – C – D – E – F – G - H
```

	A	B	C	D	E	F	G	H
A	0	1	2	3	4	5	6	7
B	1	0	1	2	3	4	5	6
C	2	1	0	1	2	3	4	5
D	3	2	1	0	1	2	3	4
E	4	3	2	1	0	1	2	3
F	5	4	3	2	1	0	1	2
G	6	5	4	3	2	1	0	1
H	7	6	5	4	3	2	1	0

	1	2	3	4	5	6	7
A	1	1	1	1	1	1	1
B	2	1	1	1	1	1	0
C	2	2	1	1	1	0	0
D	2	2	2	1	0	0	0
E	2	2	2	1	0	0	0
F	2	2	1	1	1	0	0
G	2	1	1	1	1	1	0
H	1	1	1	1	1	1	1

He tries these measures on one more graph, essentially a single linked chain of nodes, as shown in the box above. The connection matrix is shown in the center, and the length tally matrix on the right. After analyzing that graph, Nikita finally finds a difference in the “degree” and “closeness” measure, and decides to use both to analyze many more graphs.

In this example, there is a six way tie (BCDEFG) for the nodes with the greatest “degree” (0.29) of connection, but only two nodes (D and E) share the greatest “closeness” (0.44) measure. The two outermost nodes, A and H, both measure the least in “degree” (0.14) and “closeness” (0.25).

Input: Several data sets, each consisting of an initial value N, representing the number of nodes in the graph, followed on the next N lines by an NxN matrix of integer values, representing the connection matrix for the graph as described above, with single space separation.

Assumptions: The node labels for each graph will always begin with upper-case A and proceed in alpha order for as many nodes as are represented by the graph.

Output: For each data set, output four values, labeled, formatted and aligned exactly as shown below. List all nodes in alpha order that match each measure. Each output will be followed by a single line of three dashes.

Sample Input:

```
4
0 1 2 2
1 0 1 1
2 1 0 1
2 1 1 0
8
0 1 2 3 4 5 6 7
1 0 1 2 3 4 5 6
2 1 0 1 2 3 4 5
3 2 1 0 1 2 3 4
4 3 2 1 0 1 2 3
5 4 3 2 1 0 1 2
6 5 4 3 2 1 0 1
7 6 5 4 3 2 1 0
6
0 1 1 2 2 2
1 0 2 1 1 1
1 2 0 3 3 1
2 1 3 0 1 2
2 1 3 1 0 2
2 1 1 2 2 0
```

Sample Output:

```
least degree       0.33 A
greatest degree    1.00 B
least closeness    0.60 A
greatest closeness 1.00 B
---
least degree       0.14 AH
greatest degree    0.29 BCDEFG
least closeness    0.25 AH
greatest closeness 0.44 DE
---
least degree       0.40 ACDEF
greatest degree    0.80 B
least closeness    0.50 C
greatest closeness 0.83 B
---
```